

Selecting voter choices

A white paper for the EAC-NIST Human Factors Public Working Group

Center for Civic Design - Whitney Quesenbery, Dana Chisnell

University of Baltimore - Kathryn Summers

Michigan State University UARC - Sarah J. Swierenga, Graham L. Pierce, Jennifer Ismirle

JSutton Media – Jennifer Sutton

September 19, 2016

This brief white paper looks at requirements in the VVSG for how voter selections are made, comparing the current VVSG requirements with more recent research evidence and making recommendations for how the VVSG might be updated.

Table of Contents

Principles this relates to	2
Current VVSG requirement	2
Why this interaction needs guidance	3
What should the VVSG say?	3
Research evidence	4
From current voting systems	4
Hart InterCivic E-Slate	4
From voting system prototypes and research	4
Anywhere Ballot.....	4
Michigan State University Mobile Interface Specification	5
Los Angeles Voting Systems Assessment Project (VSAP)	6
Travis County STAR-Vote	6
Findings from usability testing with voters	6
What are the research gaps?	6

Principles this relates to

PRINCIPLE 2: Cast as marked

2.3: The voting system supports the voter in marking the ballot accurately

Principle 3: Marked as intended

Ballots are presented in a clear, understandable way, and is operable by all voters.

3.1: Perceivable - The default system settings for displaying the ballot work for the widest range of voters, and voters can adjust settings and preferences to meet their needs.

3.2: Operable - Voters and poll workers must be able to use all controls accurately, and all ballot changes are made with the direct control of the voter.

3.3: Understandable – Voters can understand all information as it is presented.

Current VVSG requirement

HAVA requires voting system to notify the voter of any over-votes and provide an opportunity to correct the ballot before it is cast and counted. This is reflected in the VVSG:

3.2.2 Functional capabilities

a. If the voter selects more than the allowable number of choices within a contest, the voting system shall notify the voter of the effect of this action before the ballot is cast and counted.

Discussion: In the case of manual systems, this may be achieved through appropriately placed instructions. This requirement has no force for electronic ballot interfaces, since they prevent over-voting in the first place.

3.2.2.1 Editable electronic ballot interfaces

Voting systems such as DREs and EBMs present voters with an editable interface, allowing them to easily change their votes prior to final casting of the ballot.

a. The electronic ballot interface shall prevent voters from selecting more than the allowable number of choices for each contest.

Discussion: This requirement does not specify exactly how the system must respond when a voter attempts to select an "extra" candidate. For instance, the system may prevent the selection and issue a warning, or, in the case of a single-choice contest, simply change the vote.

Why this interaction needs guidance

This issue applies only to electronic ballot marking systems.

This aspect of voting system interaction is controversial, with many different solutions to how an electronic system reacts when a voter wants to change their choice on a contest, especially if they have already marked the maximum number of choices for the contest. The system can:

- Accept the new choice and deselect (one of) the other choices.
- Refuse the new choice and force the voter to make room for the new choice by removing one of the others.
- Accept the choice, but prevent the voter from leaving the contest until the error is corrected.

However, it is worth considering, because there are several competing principles at play:

- **Consistency across voting systems.** Even though a jurisdiction may have only one electronic marking system, voters are increasingly mobile and may cross county (or state lines) several times. Inconsistency in basic operations can add to errors.
- **Consistency within a ballot.** Several projects have proposed two different interactions, depending on whether the contest allows only a single or multiple choices (also called “vote-for-one” or “vote-for-n”).
- **Keeping the voter in control.** The system should make every change to voter choices immediately visible, without forcing them to look for differences or double-check the action of the system.
- **Creating a smooth interaction that minimizes error messages.** Error messages can be confusing to voters if they are not clearly written and presented at the time the error condition is created.
- **Voter preferences vs. protecting voters.** Although there is some research evidence that voters *prefer* the system to accept the new choice, this interaction may not protect voters from errors they are not aware of.

What should the VVSG say?

Our recommendations are based on the principle:

Control of all voting selections must be under the explicit control of the voter at all times. The voting systems must neither make selections nor remove selections without direct instructions from or interaction by the voter.

This means that the voting system must inform the voter in some way when a selection is changed, especially when the change is made to an option other than the one currently in focus.

The most important objection to allowing the voting system to change a vote, even in a single-select contest, is that it opens up the possibility that the voter will not see which selection is removed, or even be clear that the first choice is now deselected.

This problem is made worse in a contest in which more than one selection can be made: which of the previous selections will be removed? The first? The most recent? The most closely

adjacent? These contests are more likely to require more than one screen to display all candidates, and therefore to have the deselection choice made off-screen.

Good instructions or messages that display the total number of choices, and dynamically reflect the number of choices left are also important.

The usability test that is part of the certification testing, the Voter Performance Protocol, needs to be updated to include changing votes in both vote-for-one and multi-select contests.

We also recommend that there be further research, working with best-practice designs tested with voters with low digital skills in a realistic setting.

Although each group designing a voting system has done research on their interface (and presumably believes that it is usable for voters), there has been no comparative research.

We don't believe that there is strong enough research data to support a design in which there are different interactions to mark a choice on different contests. Without that, human factors best practice requires a consistent and uniform interaction for all choices and contests.

This recommendation does not mean that we believe that the voter should be stopped with a "dead-end" error message. There are other design solutions that can make the message an active path forward with minimal explanation or extra interaction.

Research evidence

From current voting systems

Hart InterCivic E-Slate

The Hart system implements this interaction in two ways:

- Vote-for-one contests use alternate selection with an existing selection removed when a new choice is made.
- In multi-select contests, if the voter selects more choices than allowed, the existing choice made first is removed to make room for the new choice, without notifying the voter.

From voting system prototypes and research

Anywhere Ballot

The [Anywhere Ballot](#) tested several different approaches and messages.

The instructions above the contest indicate the total number of choices and the number of remaining choices, for example:

Vote for 1. You can vote for 1 more.

Vote for 4. You can vote for 3 more.

When all available choices have been selected, the final sentence disappears:

Vote for 3. You can vote for 3 more.
Vote for 3. You can vote for 2 more.
Vote for 3. You can vote for 1 more.
Vote for 3.

In the final version, all types of contests prevent the voter from selecting more than the allowed number of choices, using a conventional error message as a warning. Early versions of the messages included relatively lengthy (for a message) explanations, but the final message with simple directions worked well:

Uncheck the one you don't want.

Then choose the one you do want.

The modal error is not an ideal solution, however, as it stops the voter's progress and hides the list of candidates.

Michigan State University Mobile Interface Specification

MSU Usability/Accessibility Research and Consulting [created](#) and [tested](#) a user interface for accessible mobile voting systems based on prior research.

- In both vote-for-one and multi-select contests, when the maximum number of checkboxes (determined programmatically) is selected, all other checkboxes are grayed out and do not function. Deselecting a single checkbox restores all checkbox functionality.
- Only checkboxes can be used to select or deselect; candidate names and other text are not interactive, due to concerns over accidental selections or deselections by those who rest their hand or fingers on the screen as a result of physical impairments or unfamiliarity with touch screens.

Usability testing of a prototype based on the specification conducted with individuals with dexterity impairments, low vision, dyslexia, and no impairments resulted in several findings and design recommendations:

- Most users were initially confused by the need to deselect an option before selecting an alternate choice, and suggested an alert or popup explaining the deselection/selection process.

The recommendations also include guidance on the interaction details for selection that improve usability for both groups:

- All buttons should be at least 20mm in length and width, have a visible solid border that meets color contrast requirements, and include a gap of at least 6.35mm between buttons, to ensure that users do not accidentally tap the wrong one, in order to accommodate individuals with dexterity and visual impairments. Where buttons are touching, the minimum button size should be increased to provide a sufficient gap (i.e.,

two buttons directly adjacent to each other would each be at least 23mm in the relevant dimension).

- Debounce time (i.e., minimum time between user input actions, such as button-presses; additional actions taken within this window are ignored) should be at least 200ms, but not greater than 1000ms, in order to reduce user errors without increasing user frustration. Extended debounce times, large buttons, and adequate button spacing are critical for users with dexterity impairments.
- Some users with dexterity impairments had difficulties that would be resolved by allowing them to press a button longer and/or with a wider range of pressure to activate it.
- Users with dexterity impairments prefer buttons placed at the edges of the screen, to allow adequate hand support.

Los Angeles Voting Systems Assessment Project (VSAP)

The VSAP prototypes started with accepting the new choice and deselect (one of) the other choices in all contests. The final prototype has a mixed approach:

- Vote-for-one contests use alternate selection.
- In multi-select contests, when the voter makes the last available choice, the other options are greyed out and disabled (in a rapid animation) until a choice is deselected. There is no message or other indication of how to make a disabled option active.

Travis County STAR-Vote

The current RFI outlining the requirements for STAR-Vote requires a “toggle design” for single selection, with an interface in which voters have to explicitly deselect a candidate for multi-select contests

- 9.3.1.10.1- For races in which only a single selection may be made, after a candidate is selected, the user must be able to directly select a different candidate without having to deselect the initial selection. If a user wanted to deselect a selected candidate (e.g. so they can make ‘no selection’), they should be able to simply touch the candidate again (a toggle design);
- 9.3.1.10.2- For races in which more than one selection may be made, ‘pop-up’ instructions must be displayed if the voter attempts to make more than the allowed number of selections explaining how the voter may change their selections;

Findings from usability testing with voters

Low literacy voters are somewhat likely to see every new screen as a new contest, which is not the case when the candidate list spreads across two screens. If voters feel prompted to make a choice on every screen, these voters would be changing their votes without knowing it.

What are the research gaps?

It can be difficult to test error conditions in small qualitative testing and with instructed voting but a comparative test of the three possible interaction models (always make the change, never

make the change without confirmation, make the change in vote-for-one contests, but not in multi-select contests) is needed.

This might also be investigated through a comparative analysis of data from elections using different existing systems, if there was a way to detect problems through system logs or other data.

There also needs to be more design work on alternative ways to implement a requirement that all changes be confirmed so that the interaction is helpful in ensuring voter intent, but not annoying to voters, finding a balance between protecting voters and minimizing error interactions.

This issue also requires some top-level design research to come up with designs that address these issues:

- What notification is sufficiently informative without being annoying?
- What tools serve to attract visual attention and how can they be approximated in a non-visual way?
- What other alternatives could there be to an “error message” such as a message that presented the current selections plus the desired addition, allowing the voter to make a choice among this list?